UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/822,535	03/30/2001	Erik S. Ruf	MS 160347.2/40062.120US01	1094

7590 04/08/2004

Homer L. Knearl
Merchant & Gould P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903

EXAMINER	
RUTTEN, JAMES D	

ART UNIT	PAPER NUMBER
2122	3

DATE MAILED: 04/08/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/822,535	Applicant(s) RUF, ERIK S.	
	Examiner J. Derek Rutten	Art Unit 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 March 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23* is/are rejected.

7)☒ Claim(s) *10 and 23* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *30 March 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 2.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-23 have been examined.

Drawings

2. Figures 1 and 8 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

3. Claims 10 and 23 are objected to because of the following informalities: There appears to be a typo in line 8 of claim 10, and line 6 of claim 23, resulting in the omission of the word "the". The phrase "restricted set" should be --the restricted set--. Appropriate correction is required.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claim 17 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

6. Claim 17 recites the limitation "the computer process" in line 1. There is insufficient antecedent basis for this limitation in the claim. For the purpose of further examination, this limitation will be interpreted to refer to --the method--.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claim 1-23 rejected under 35 U.S.C. 103(a) as being unpatentable over "C++: Effective Object-Oriented Software Construction" by Dattatri (hereinafter referred to as "Dattatri"), in view of "Compiler Transformations for High-Performance Computing" by Bacon et al. (hereinafter referred to as "Bacon").

As per claim 1, Dattatri discloses:

generating a table having an entry associated with the target method of the receiver object (bottom of page 705: "Virtual functions are implemented through a table of pointers to functions, the vtbl."); *and*

generating an optimized instruction in association with the call site to retrieve a return value associated with the target method (top of page 707: "`This (pb->g()) can be written as ( *(pb->vptr[2]) )();`").

Dattatri does not expressly disclose the use of a return constant table encoded on computer program product.

However, in an analogous environment, Bacon teaches the construction of a cache to store the constant results of a side effect free procedure (pages 391-392, Section 6.8.9: "In such cases, it is possible to cache the results of recent invocations.").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacon's return constant cache table storage. One of ordinary skill would have been motivated to eliminate a method call in favor of directly accessing a constant value from a cache table. Method calls generate many extra instructions and computations, whereas a table lookup simply loads the value from memory. Further, it would have been obvious to one of ordinary skill in the art at the time the invention was made to encode the computer programs of Dattatri and Bacon on a computer program product. One of ordinary skill would have been motivated to encode a computer program as a means to distribute the program to a customer.

As per claim 2, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose inserting a constant return value.

However, Bacon teaches the insertion of a constant return value in a cache (pages 391-392, Section 6.8.9 as referenced above).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bacon's method of constant value insertion into Dattatri's method table. One of ordinary skill would have been motivated to store a constant return value in a method table which provides fast access and to avoid having to execute a costly method call.

As per claim 3, the above rejection of claim 1 is incorporated. Dattatri discloses a dispatch table (bottom of page 705 as referenced above). Dattatri does not expressly disclose inserting a return value in a separate constant return table.

However, Bacon teaches the insertion of a constant return value in a separate cache (pages 391-392, Section 6.8.9 as referenced above).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bacon's method of constant value insertion into Dattatri's method table. One of ordinary skill would have been motivated to store a constant return value in a separate table in order to maintain a simple data layout which also provides fast access while avoiding having to execute a costly method call.

As per claim 4, the above rejection of claim 3 is incorporated. Dattatri does not expressly disclose associating the return constant table with a receiver object.

However, Bacon teaches using a cached result with a procedure (page 391 Section 6.8.9 and page 392 Figure 51).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to associate Bacon's cache table with Dattatri's methods. One of ordinary skill would have been motivated to provide a means for accessing an improved data structure from a method that could benefit from using it.

As per claim 5, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose the determination of a non-transformable site and inserting into a dispatch table.

However, Bacon teaches the determination of a procedure as being non-transformable (page 391, Section 6.8.9, paragraph 1: "side-effect free"), and insertion of a return value into a cache (page 391 Section 6.8.9, paragraph 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacon's non-transformable determination and insertion of return values in a cache. One of ordinary skill would have been motivated to ensure that the elimination of a method call was only made to a method that did not change the state of the program. Otherwise, the elimination of a method call would also eliminate whatever changes to state that would have occurred in the method.

As per claim 6, the above rejection of claim 1 is incorporated. Further, Dattatri discloses evaluating a plurality of possible target methods (bottom of page 705). Dattatri does not expressly disclose identification of return values.

However, Bacon teaches the identification of constant return values (pages 391-392, Section 6.8.9).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacons identification of return values. One of ordinary skill would have been motivated to identify as many return values as possible in order to maximize the benefit of replacing a method call with access to stored values.

As per claim 7, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose the determination of side-effects.

However, Bacon teaches the application of function memoization only in the case when there are no side-effects (page 391 Section 6.8.9 paragraph 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacon's side effect determination. One of ordinary skill would have been motivated to ensure that the elimination of a method call was only made to a method that did not change the state of the program. Otherwise, the elimination of a method call would also eliminate whatever changes to state that would have occurred in the method.

As per claim 8, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose an instruction for retrieval from the constant table.

However, Bacon teaches the generation of fetching instructions for retrieval of a return value from a data structure (page 392 Figure 51).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacon's fetching instructions. One of ordinary skill would have been motivated to access return values stored in a table. Access to a table is generally faster than invoking a method call, and without instructions to access the table, it would be difficult to use.

As per claim 9, the above rejection of claim 1 is incorporated. Further, Dattatri discloses evaluating a plurality of possible target methods (bottom of page 705). Dattatri does not expressly disclose identification and storage of return values.

However, Bacon teaches the identification and storage of return values (pages 391-392, Section 6.8.9).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacons identification and storage of return values. One of ordinary skill would have been motivated to identify as many return values as possible in order to maximize the benefit of replacing a method call with access to stored values.

As per claim 10, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose identification and optimization of control variables.

However, Bacon teaches constant propagation including:

identifying a restricted set of one or more values of a control variable associated with a control operation (page 380 Figure 35: "n=64");

identifying a restricted set of one or more types associated with the restricted set of one or more values of the control variable (Compilers inherently identify the types of variables used in the programs that they analyze. Without type identification, a compiler might attempt calculations with two incompatible pieces of data.); *and*

optimizing one or more control targets associated with the control operation based on the restricted set of one or more types (pages 379 and 380, Section 6.6.1 and Figure 35).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bacon's constant propagation in Dattatri's object model. One of ordinary skill would have been motivated to precompute variables in a program in order to reduce computation at runtime, and also to reveal opportunities for other optimizations such as loop optimization.

As per claim 11, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose identification, mapping and optimization of control variables.

However, Bacon teaches constant propagation including:

identifying a restricted set of one or more values associated with a control variable (page 380 Figure 35: "n=64");

identifying one or more target methods providing the values associated with the restricted set (page 380 Figure 35);

mapping between the restricted set of values of the control variable and a restricted set of types based on the one or more target methods (Mapping values of variables and types or methods is an inherent operation of compilers. Without mapping, a compiler might attempt calculations with two incompatible pieces of data.)*; and*

optimizing one or more control targets associated with the control statement based on the restricted set of types (pages 379 and 380, Section 6.6.1 and Figure 35).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bacon's constant propagation in Dattatri's object model. One of ordinary skill would have been motivated to precompute variables in a program in order to reduce computation at runtime, and also to reveal opportunities for other optimizations such as loop optimization.

As per claim 12, Dattatri discloses:

A method (page 701: "How Objects are Represented in C++") *comprising:*

generating a table associated with the receiver object, the table having an entry associated with the target method of the receiver object (bottom of page 705: "Virtual functions are implemented through a table of pointers to functions, the vtbl.")*; and*

generating an optimized instruction in association with the call site to retrieve a value associated with the target method (top of page 707: "`This (pb->g()) can be written as ( *(pb->vptr[2]) )();`").

Dattatri does not expressly disclose the determination of side effects or the use of a return constant table.

However, Bacon teaches the application of function memoization only in the case when there are no side-effects (page 391 Section 6.8.9 paragraph 1: "Memoization is an optimization that is applied to side-effect free procedures (that is, procedures that do not change the state of the program, also called *referentially transparent*)."). Bacon also teaches the construction of a cache to store the constant results of a side effect free procedure (pages 391-392, Section 6.8.9: "In such cases, it is possible to cache the results of recent invocations.").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacon's side effect free constant storage. One of ordinary skill would have been motivated to ensure that the elimination of a method call was only made to a method that did not change the state of the program. Otherwise, the elimination of a method call would also eliminate whatever changes to state that would have occurred in the method. It is also beneficial to eliminate a method call in favor of directly accessing a constant value from memory. Method calls generate many extra instructions and computations, whereas a table lookup simply loads the value from memory.

As per claims 13-21, the above rejection of claim 12 is incorporated. Further, all other limitations have been addressed in the above rejections of claims 2-4 and 6-11, respectively.

As per claims 22 and 23, all limitations have been addressed in the above rejections of claims 1 and 10, respectively.

Conclusion

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent 6,161,217 to Detlefs et al. discloses a method for determining what type is most likely to be appropriate for a virtual method call, and optimizing the method call using that type. Method inlining can be used with constant propagation for program optimization.

"The Direct Cost of Virtual Function Calls in C++" by Driesen et al. discusses virtual function tables (vft) and the implementation of thunks which, in the case of single inheritance, point directly to the target function instead of having to calculate the address using an offset stored in the vft.

"Optimizing Dynamically-Typed Object-Oriented Languages with Polymorphic Inline Caches" by Holzle et al. discusses caching multiple targets per call site. Holzle also discusses the inlining of short methods such as a simple "get" method which might return a constant.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (703) 605-5233. The examiner can normally be reached on M-F 6:30-3:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (703) 305-4552. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jdr

TUAN DAM
SUPERVISORY PATENT EXAMINER